УДК 004.056.53

Луцький М.Г., д.т.н.

ОПТИМАЛЬНИЙ ГРАФІЧНИЙ ФОРМАТ СТЕГАНОЗАХИСТУ

Національний авіаційний університет

Визначено оптимальніший графічний формат файлу для реалізації стеганографічної системи. Сформовано графічні залежності відхилень структурних ознак контейнера-оригінала від контейнера-результату

Вступ

Завдання надійного захисту інформації від несанкціонованого доступу є однією з найактуальніших проблем сучасного суспільства. Способи та методи приховування повідомлень відомі з давніх часів, дана сфера людської діяльності отримала назву стеганографія. Внаслідок всебічного використання комп'ютерних технологій у сучасному суспільстві з'явився новий напрямок в області захисту інформації - цифрова стеганографія.

Цифрова стеганографії - напрям класичної стеганографії, заснований на приховуванні додаткової інформації в цифрових об'єктах, спричиняючи при цьому деякі спотворення цих об'єктів. Але, як правило, дані об'єкти є мультимедійними (зображення, відео, аудіо) та внесення спотворень, які знаходяться нижче порогу чутливості середньостатистичної людини не призводять до помітних видозмін цих об'єктів.Використання методів цифрової стеганографії для реалізації захисту інформації призводить до створення стеганографічної системи. Під стеганографічною системою слід розуміти об'єднання методів і засобів, які використовуються для створення прихованого каналу передачі інформації.

Постановка задачі

Стеганографічна система виконує вбудовування повідомлення в контейнер, передавання заповненого контейнера стеганоканалом та декодування приховуваного повідомлення. В якості фіксованого контейнера для передачі прихованої інформації частіше використовують растрові зображення. При внесенні повідомлення до контейнера відбувається його певне спотворення. Для підвищення стійкості стеганографічної системи слід прагнути якнайменше змінювати показники структурних ознак контейнера. До них відносяться: роздільна здатність растрового зображення, розмір растру, глибина кольору зображення.

Зберігання растрового зображення (стеганоконтейнера) відбувається у вигляді певного файлу. Кожен файл зображення має свій формат, в залежності від алгоритму зберігання графічної інформації. В залежності від формату, при внесенні повідомлення до зображення (контейнера), можлива зміна розміру файлу.

Метою даної статті є аналіз сучасних графічних форматів растрового зображення в умовах реалізації процесів стеганозахисту. У даній статті на базі аналізу буде визначено оптимальніший графічний формат файлу для реалізації стеганографічної системи. Буде сформовано графічні залежності відхилень структурних ознак контейнера-оригінала від контейнера-результату.

Вирішення поставленого завдання

Зберігання растрових зображень відбувається у вигляді певних файлів. На теперішній час відомо багато форматів файлів для растрових зображень. Растрове зображення зберігається в стиснутому вигляді. В залежності від алгоритмів стиснення може бути можливим чи неможливим відновлення зображення в точності таким самим, яким воно було до стиснення. На рис. 1 зображені найпоширеніші формати файлів растрового зображення. ***BMP*-формат файлу.** З даним форматом працює велика кількість програм, через те, що його підтримка інтегрована в ОС *Windows*.

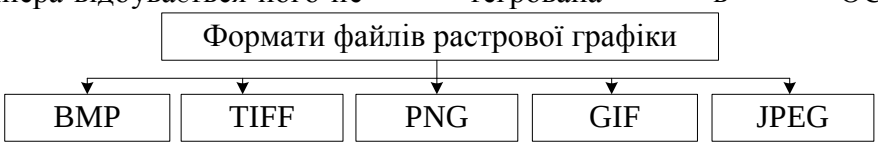

Рис. 1. Найпоширеніші формати файлів растрового зображення

Глибина кольору в *BMP* форматі може становити 2-48 біт на піксель, при максимальних розмірах растру – 65535×65535. *BMP* – файл містить 4 частин.

Заголовок *BITMAPFILEHEADER* містить загальний опис файлу, на який відводиться 14 байт (рис. 2). Далі розташований заголовок зображення – *BITMAPINFOHEADER*, в якому зберігається опис розмірів растру та колірного формату пікселів. На нього розраховано – 40 байт. Наступна складова – палітра (*RGBQUAD*). Розмір палітри залежить від кількості кольорів. У деяких випадках палітра може бути відсутньою. Після палітри, в файлі *BMP*–формату, записується растр у вигляді масиву.

Рис. 2. Структура BMP-файлу

Кількість байт, у масиві, визначається розмірами растру та кількістю біт на піксель. У форматі *BMP* підтримується стиснення без втрат за алгоритмом *RLE* (рис. 3).

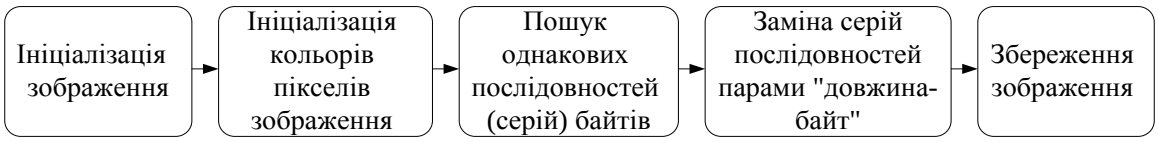

Рис. 3. Структурна схема реалізації алгоритму RLE

Цей алгоритм базується на такому принципі: заміна повторюваних групи елементів вихідної послідовності на пару (“довжина-символ”), або тільки на довжину. Стиснення в *RLE* відбувається за рахунок послідовності однакових байт в початковому зображенні. Заміна їх на пари зменшує надлишковість даних.

***GIF*–формат.** *GIF* – це растровий графічний формат, який використовує до 256-и індексованих кольорів із 24-х бітного діапазону *RGB*. Зображення такого формату відбувається порядкове зберіганням, з підтримкою тільки формату з визначеною палітрою кольорів.

У *GIF*-форматі використовується стиснення за алгоритмом *LZW* (рис. 4), який в поєднанні із індексованими різними кольорами робить даний формат ідеальним при зберігання і передачі зображення з малою кількістю кольорів. Алгоритм стиснення *LZW* відноситься до форматів стиснення без втрат. Тобто, при відновлені з *GIF*, дані з точністю відповідатимуть оригіналу.

При такому стисненні відбувається пошук повторюваних комбінацій різних кольорів ("фраз"), які записуються у вигляді ключів. Для кодування зображення використовуються вже створені ключі. Цей метод досконаліший за *RLE* при роботі з областями, що мають переходи кольорів, однак кодування в *LZW* вимагає більше системних ресурсів.

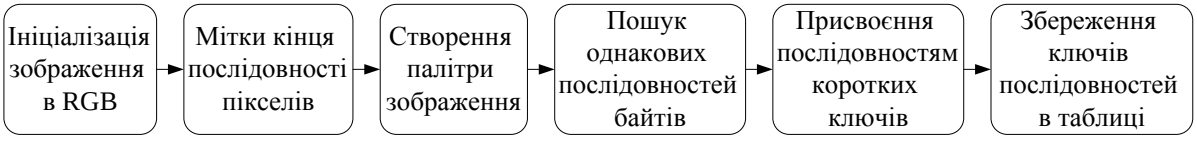

Рис. 4. Структурна схема реалізації алгоритму LZW

Таким чином, добре стискаються зображення, рядки яких мають повторювані ділянки. Формат *GIF* дозволяє здійснювати черезстрокове зберігання даних. При цьому рядки розбиваються на групи, і змінюється порядок зберігання рядків у файлі. При завантаженні, зображення проявляється поступово. Завдяки цьому, маючи тільки частину файлу, можна побачити зображення цілком, але з меншим дозволом.

Особливістю *GIF*-формату є підтримка анімаційних зображень. Вони визначені у вигляді послідовності з декількох статичних кадрів, а також інформацією про те, скільки часу кожен кадр повинен бути відображений на екрані.

***PNG*-формат.** Формат *PNG* спроектований для заміни застарілого і більш простого формату *GIF*. *PNG* - растровий формат збереження графічної інформації, що використовує стиснення без втрат. Формат *PNG* був розрахований, перш за все, для використання в Інтернеті і редагування графіки. Даний формат підтримує растрові зображення з глибиною кольору 16, 24 та 48 біт.

Цей формат характеризується більш сильнішим рівнем стиснення для файлів з великою кількістю кольорів ніж *GIF*. *PNG* використовує стиснення без втрат за алгоритмом *Deflate*. Даний алгоритм використовує комбінацію алгоритму *LZ77* і алгоритму Хаффмана.

Принцип алгоритму *LZ77* будується на принципі ковзаючого вікна і механізмі кодування збігів (рис. 5). Метод кодування згідно з принципом ковзаючого вікна враховує вже раніше переглянуту частину інформації, яку використовує як словник. Він намагається замінити черговий фрагмент повідомлення на покажчик у вміст словника. Ковзаюче вікно можна представити у вигляді буфера, який організований так, щоб запам'ятовувати переглянуту раніше інформацію і надавати до неї доступ. Таким чином, друге і наступне виникнення збігів однакових послідовностей кольорів пікселів замінюються посиланням на їх першу появу.

При виникненні збігів, вони кодуються парою: довжина збігу та зміщення. Кодована пара трактується як команда копіювання символів, довжиною збігу, з позиції ковзаючого вікна, що задається зміщенням. Використання кодованої пари довжина-зміщення є ефективним у випадку, коли значення довжини перевищує значення зміщення.

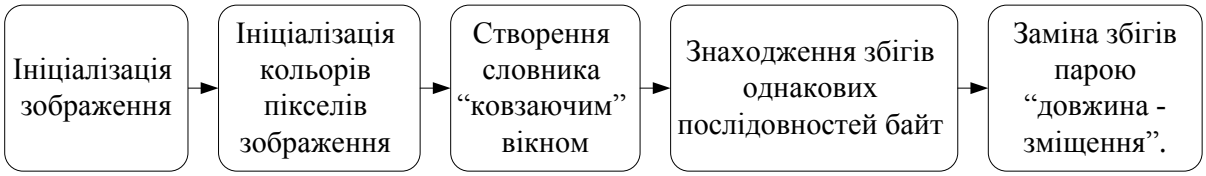

Рис. 5. Схема реалізації алгоритму *LZ77*

Алгоритм Хаффмана використовує частоту появи однакових байт в зображенні (рис. 6). Зіставляє символам вхідного потоку, які зустрічаються більшу кількість разів, послідовність біт меншої довжини та навпаки. Першим етапом алгоритму є прочитання файлу зображення та підрахування частоти появи кожного байту кольору. Потім створюємо таблицю відповідності байту кольору його частоті та впорядковуємо за спаданням. Наступним етапом є побудова дерева за допомогою створених вузлів таблиці. Після створення дерева відбувається кодування файлу та збереження декодую чого дерева. Алгоритм Хаффмана вимагає читати вхідний файл двічі, один раз підраховуючи частоти появи байтів, іншим разом виконуючи безпосередньо кодування.

Закодовані у відповідності з форматом *Deflate* дані представляють собою набір блоків, порядок яких збігається з послідовністю відповідних блоків вихідних даних. Довжина блоків першого типу не може перевищувати 64 кбайт. Кожен блок другого та третього типу складається з двох частин: описи двох таблиць кодів Хаффмана.

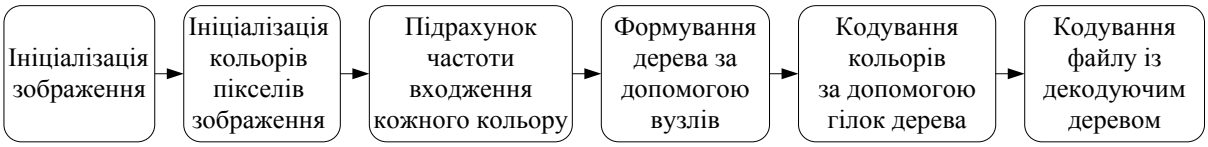

Рис. 6. Схема реалізації алгоритму Хаффмана

Для пошуку фраз використовується метод хеш-ланцюгів. Хеш-функція обчислюється на підставі трьох байтів даних. На кожному кроці компресор читає черговий 3-байтовий рядок, що розташовується на початку буфера. Хеш-ланцюжок аналізується з метою знаходження найдовших збігів між буфером і фразами, на які посилаються елементи (вузли) хеш-ланцюжка. Оновлення хеш-ланцюжків організовано таким чином, що пошук починається з нових вузлів, що дозволяє змістити розподіл частот зсувів кодованих фраз на користь коротких зміщень і, отже, поліпшити стискання, тому що невеликі зсуви мають коди малої довжини. Для прискорення кодування в разі обробки надлишкових даних дуже довгі хеш-ланцюжки скорочуються до певної довжини, що задається параметрами алгоритму. Скорочення проводиться залежно від довжини вже знайдених збігів: чим вони довші, тим більше скорочуємо.

***JPEG*-формат.** *JPEG* - растровий формат збереження графічної інформації, який використовує стиснення із втратами. Даний алгоритм стиснення оптимальний для фотографій та зображень, які містять реалістичні дані із поступовими змінами яскравості та кольору. Найбільшого застосування, цей формат отримав у цифровій фотографії і передачі графічної інформації мережею Інтернет.

При стисненні, зображення перетворюється з колірного простору *RGB* в *YUV* (рис.7). Після перетворення, для каналів зображення *U* і *V*, що відповідають за колір, може виконуватися "проріджування ", яке полягає в тому, що кожного блоку з 4 пікселів яскравого каналу *Y* ста

вляться у відповідність усереднені значення *U* і *V*. При цьому для кожного блоку 2x2 замість 12 значень (4 *Y*, 4 *U* і 4 *V*) використовується всього 6 (4 *Y* і по одному усередненому *U* і *V*). Далі компонент *Y* та відповідаючі за колір компоненти *U* і *V* розбиваються на блоки 8x8 пікселів. Кожен такий блок піддається дискретному косинусному перетворенню (ДКП). Отримані коефіцієнти ДКП квантуються і пакуються з використанням кодів Хаффмана.

Рис. 7. Структурна схема реалізації алгоритму *JPEG*

Матриці, які використовуються для квантування коефіцієнтів ДКП, зберігаються в заголовній частині *JPEG*-файлу. Зазвичай вони будуються так, що високочастотні коефіцієнти піддаються більш сильному квантуванню, ніж низькочастотні. Це призводить до меншої деталізації дрібних деталей на зображенні. Чим вище ступінь стиснення, тим більш сильному квантуванню піддаються всі коефіцієнти.

***TIFF*-формат.** *TIFF* - формат зберігання растрових графічних зображень . *TIFF* набув широкого застосування при зберіганні зображень з великою глибиною кольору. Завдяки своїй сумісності з більшістю професійного ПЗ для обробки зображень, формат *TIFF* дуже зручний при перенесенні зображень між комп'ютерами різних типів (наприклад, з PC на *Mac*). Структура формату дозволяє зберігати зображення з палітрою, а також в різних колірних просторах. *TIFF* має можливість зберігати зображення із стисненням та без стиснення. Ступень стиснення залежить від особливостей самого зберігання зображення, а також від алгоритму. Формат *TIFF* дозволяє використовувати такі алгоритми стиснення: *RLE*, *LZW* та *JPEG*. При цьому *JPEG* є просто інкапсуляцією формату *JPEG* у формат *TIFF*. Формат *TIFF* дозволяє зберігати зображення, по стандарту *JPEG*, без втрат даних (*JPEG-LS*).

Таким чином, формат графічного файлу вказує на спосіб стиснення растрового зображення. Так, стиснення без втрат виконується при зберігання зображення у *PNG*, *GIF* та *BMP*-формату (зменшення надлишковості даних). Стиснення з втратами відбувається у *JPEG*-формату (відкидається частина інформації). *TIFF*-формат підтримує стиснення із втратами та без втрат.

Для проведення оцінки зміни розміру файлу контейнера-оригінала та контейнера-результату, була використана синя складова статичного 24-бітового *RGB* зображення із розміром растру – 240x240.

Растрове зображення являє собою матрицю пікселів. Певний піксель може зберігати інформацію про один певний колір. Кожен піксель кольорового зображення можна розкласти на колірні компоненти *R*, *G* і *B*, в колірній схемі *RGB* (рис. 8). Внаслідок цього, зображення розкладається на три колірні компоненти, що розміщуються одна за одною у спільному масиві. Для обробки зображення, колірні характеристики зображення переводиться у числову матрицю.

Приховування інформації можна проводити у всі три колірні компоненти. Зміна у кожному окремому колірному компоненті може призвести до різних візуальних спотворень результуючого зображення. Око людини найбільшу чутливість має до спектру червоного кольору, трохи слабкішу – до зеленого та істотно слабкішу – до синього. Оскільки людина має найслабкішу чутливість до синього спектру, то зміни в колірній компоненті синього кольору будуть найменше помітні. Тому приховування інформації оптимальніше проводити у синю компоненту.

При кодуванні пікселя кольорового зображення використовується 24 біти, по 8 бітів на кожну колірну компоненту. Глибина кольору компоненти становить 8 біт.

Найменше значущий біт несе менше всього інформації. Людина зазвичай не здатна помітити зміну в цьому біті. Тому його можна використовувати для вбудовування прихованої інформації, шляхом заміни найменш значущих бітів пікселів зображення бітами повідомлення.

Для проведення визначення оптимального формату файлу розмір місця синьої складової, що модифікувався, складав від 20% до 100% (рис. 8). Модифікування виконувалось методом заміни найменше значущого біту елементу зображення (рис. 9).

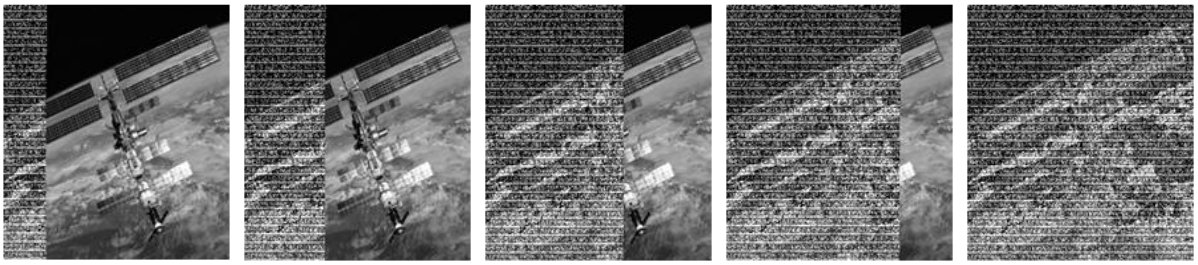

Рис. 8. Візуальне відтворення ступеню заповнення синьої складової

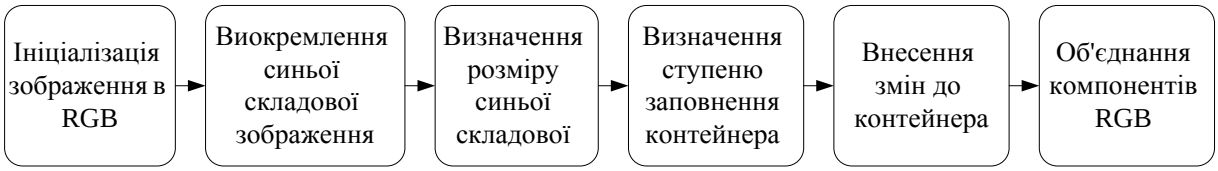

Рис. 9. Структурна схема реалізації приховування інформації в компоненту синього кольору

При знаходженні оптимального формату файлу для стеганографічного перетворення виконаємо порівняння отриманих зображень на основі показника зміни розміру файлу. Результати розмірів файлів після стеганоперетворення в залежності від ступеня заповнення наведені у таблиці 1.Таким чином, за даним показником оцінювання, оптимальнішим є BMP-формат.

Таблиця 1. Розмір файлу

№	Формат файлу	Ступінь заповнення контейнера, %					
		0 %	20%	40%	60%	80%	100%
		Розмір файлу після стеганоперетворення, байт					
1	***TIFF***	202 200	202 208	202 208	202 216	202 208	202 212
2	***BMP***	172 854	172 854	172 854	172 854	172 854	172 854
3	***PNG***	108 181	108 462	108 576	108 653	108 718	108 806
4	***JPEG***	47 015	47 016	47 019	47 027	47 011	47 017
5	***GIF***	40 550	40 619	40 620	40 562	40 580	40 636

Розмір даного файлу не змінюється. Зміна розмірів файлів інших форматів обумовлена алгоритмом стиснення інформаційної послідовності, що використовуються у даних форматах. Менш надійнішим є *JPEG*-формат, різниця розміру заповненого зображення та зображення-оригіналу становила від 1 до 12 байтів. При використанні *TIFF*, різниця розміру файлів змінювався від 8 до 16 байтів. При *GIF* – від 12 до 70 байтів, а *PNG* – 354 до 625 байтів.

Висновок

Отже, на основі проведеного дослідження, можливо визначити оптимальніший та найстійкіший формат графічного файлу до стеганографічних перетворень. За результатом досліджень, найоптимальнішим є *BMP*-формат. При внесенні прихованих повідомлень зберігається розмір файлу даного формату, що підвищує стійкість стеганосистеми до стеганоаналізу. Останні формати файлів є менш стійкими до стеганографічний перетворень. Це зумовлено змінами у розмірі файлів, що підвищує шанси порушника до виявлення стеганоканалу.

Список літератури

1. Грибунин В. Г. Цифровая стеганография/ В. Г.Грибунин, И. Н.Оков, И. В.Туринцев. – М.: Солон-Пресс, 2002. – 272 с.

2. Конахович Г.Ф. Компьютерная стеганография: Теория и практика/ Г.Ф.Конахович, А.Ю.Пузыренко. – К.: МК-Пресс, 2006. – 288 с.

3. Юдін О. К. Захист інформації в мережах передачі даних: підруч. / Г.Ф.Конахович, О.Г.Корченко, О.К.Юдін. — К.: Вид-во ТОВ НВП «ІНТЕРСЕРВІС», 2009. — 714 с.